Visible Light Communication Networks Based on Linux-Enabled Light Bulbs

Stefan Schmid

http://www.disneyresearch.com/project/visible-light-communication

Outline

Visible Light Communication

The Visible Light Spectrum as a Communication Medium

Building Blocks

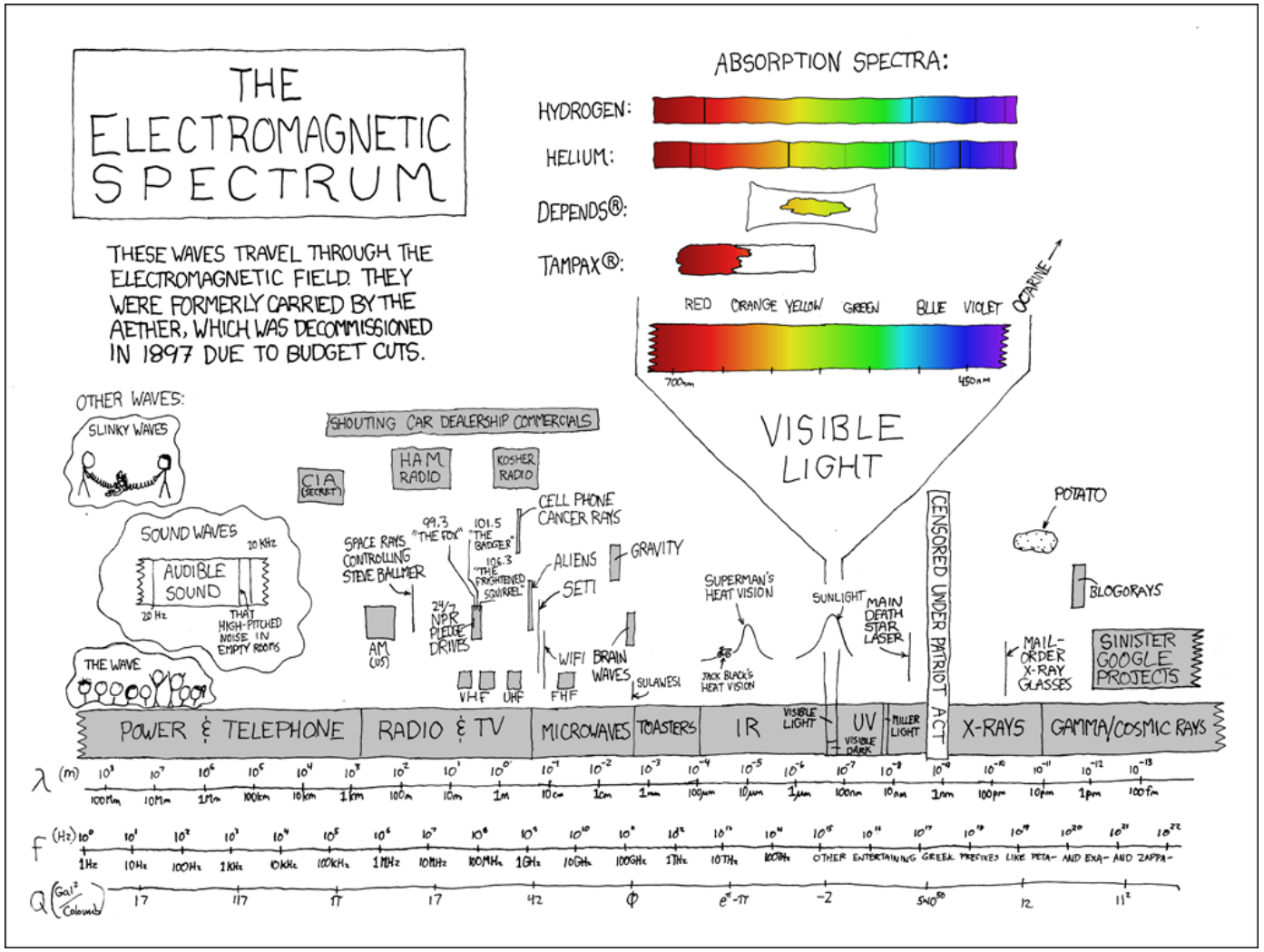
Bringing Low-Cost Visible Light Communication to Everyday Devices

Linux Light Bulbs

Linux Network Stack Integration and Evaluation

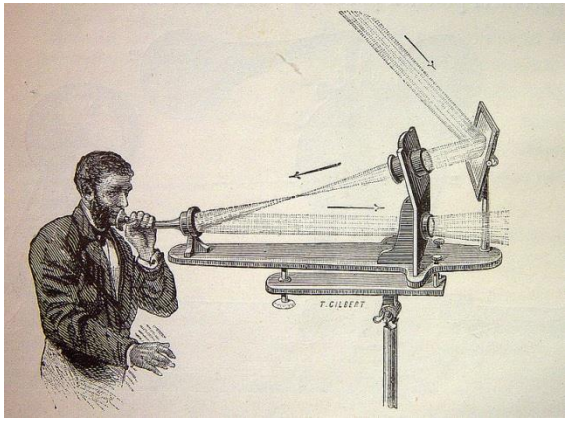
Visible Light Communication

Visible Light Spectrum

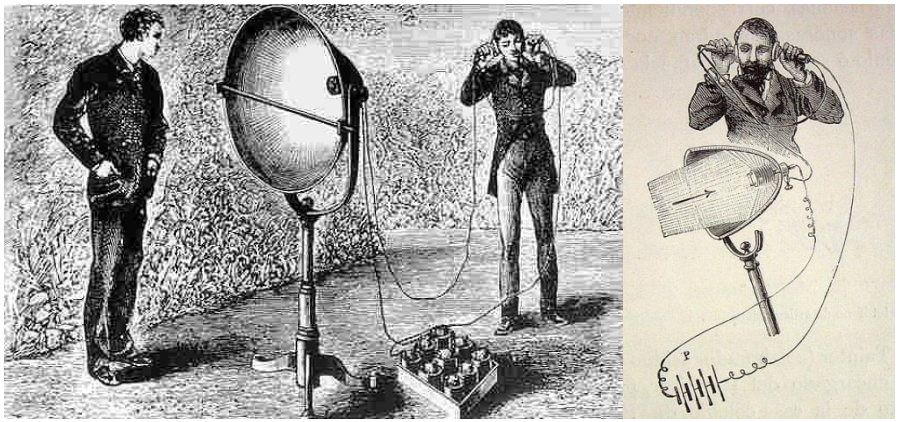


[xkcd: https://xkcd.com/273/]

Photophone (Graham Bell, 1880)



Transmitter

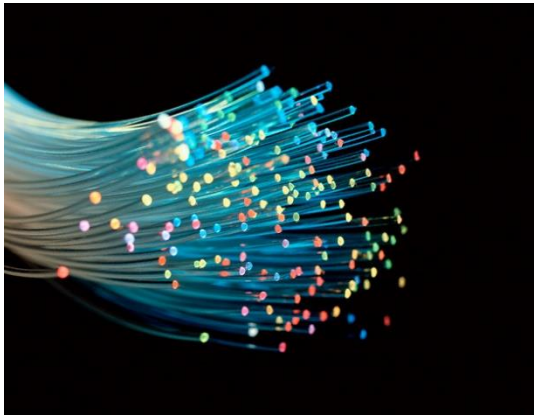


Receiver

- First dated application of Visible Light Communication
- First wirelessly delivered telephone message (200m)

[Wikipedia: https://en.wikipedia.org/wiki/Photophone)]

Visible Light Communication - Today



Fiber optics (wired)

Li-Fi Consortium

Free-space laser communication

LED-to-LED communication

pureLiFi

- 5 Mb/s symmetric link
- VLC and IR

[pureLiFi: http://purelifi.com/lifi-products/li-1st/]

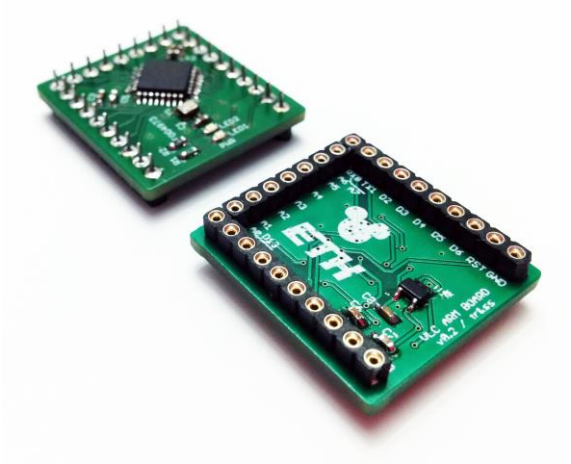
Fraunhofer HHI

- 500 Mb/s over 4 m
- 120 Mb/s over 20 m

[Fraunhofer HHI: http://www.hhi.fraunhofer.de/departments/photonic-networks-and-systems/products-and-services/vlc-system-500-mbits.html]

Building Blocks

Low-Cost Software-Defined VLC

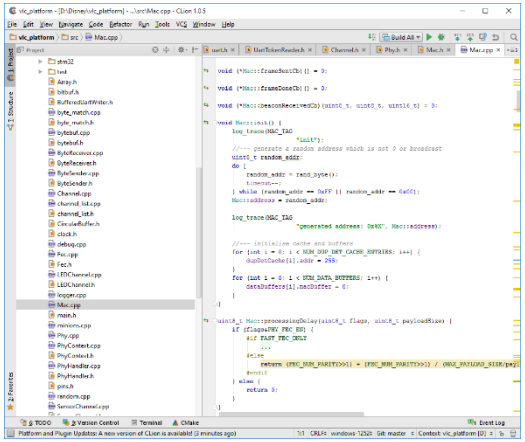


Microcontroller

+

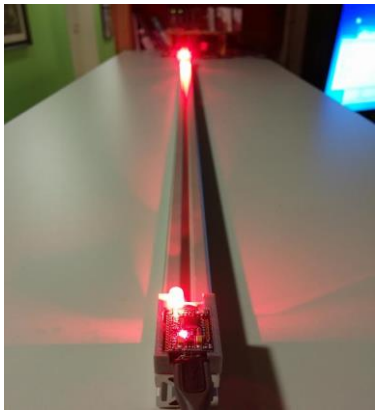
Sender/Receiver

+



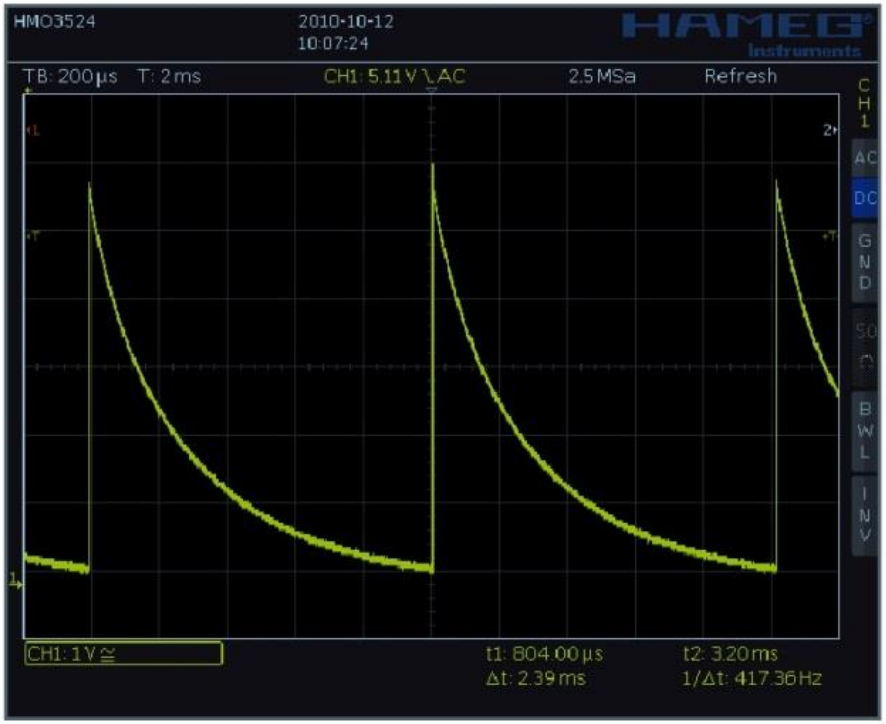
Software

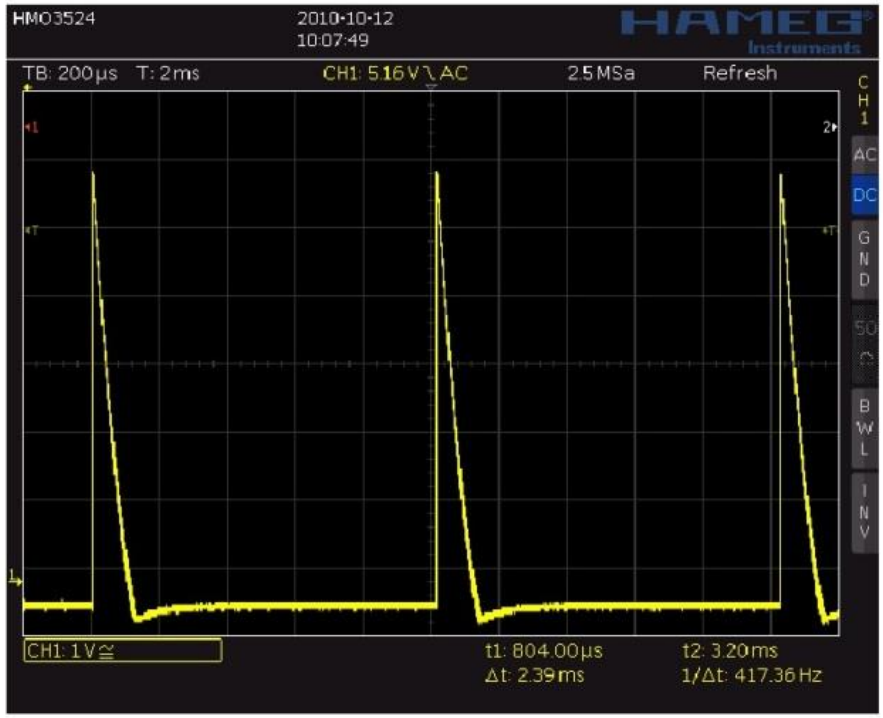
=



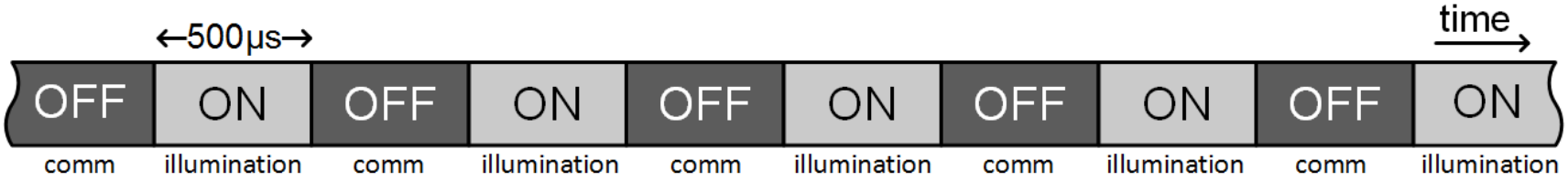
VLC communication link:
1 kb/s over 2 meters

Light Sensing with LEDs





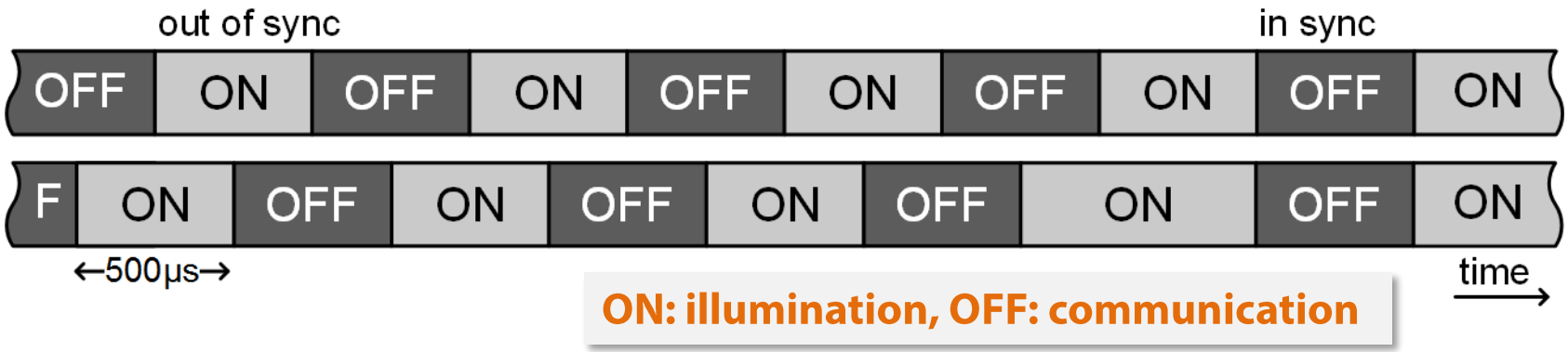
Illumination & Communication



ON: illumination, OFF: communication

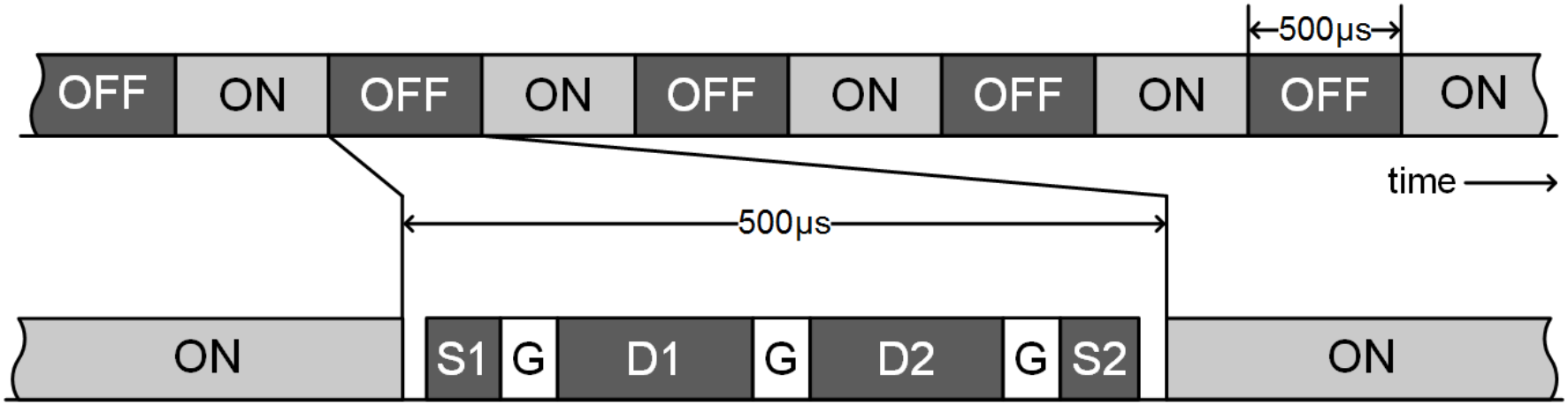
- Combining illumination and communication
- Periodic idle pattern
- ON slot -> illumination, emitting light
- OFF slot -> communication, data modulation, sensing

Synchronization



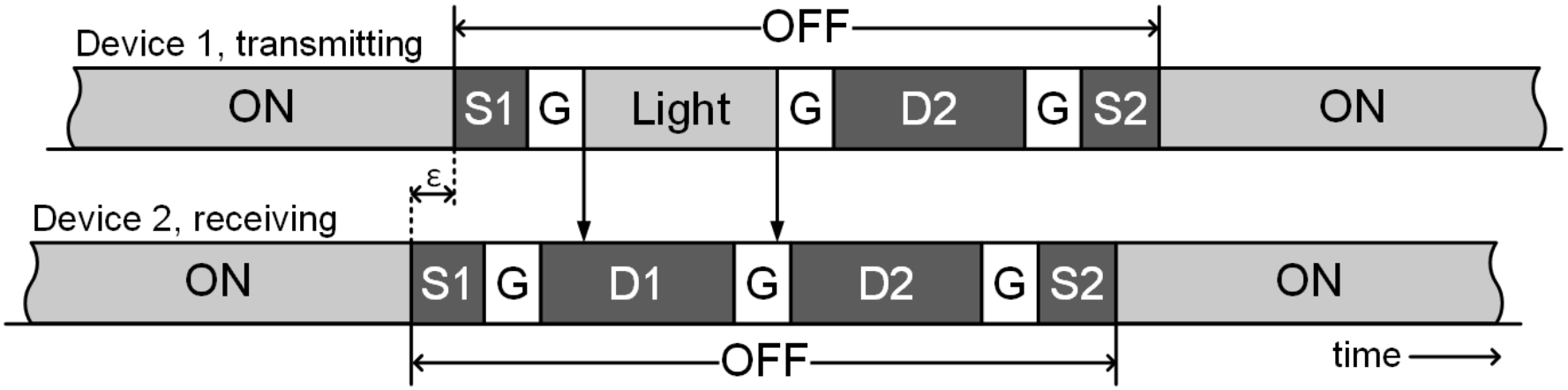
- Initial synchronization (startup offset)
- Continuous synchronization - also during RX/TX (clock drifts, long packet duration)

Slot Structure



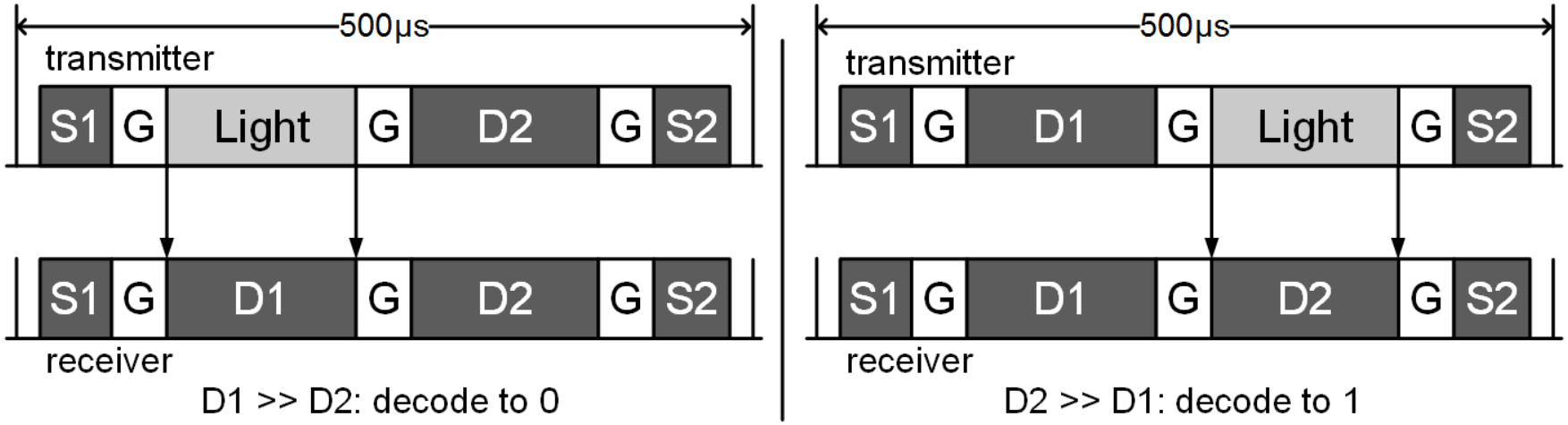
- Multiple (different) measurement intervals during OFF slot
- Separation of illumination and data modulation
- Continuous synchronization (also during data transmission)

Synchronization & Guard Intervals



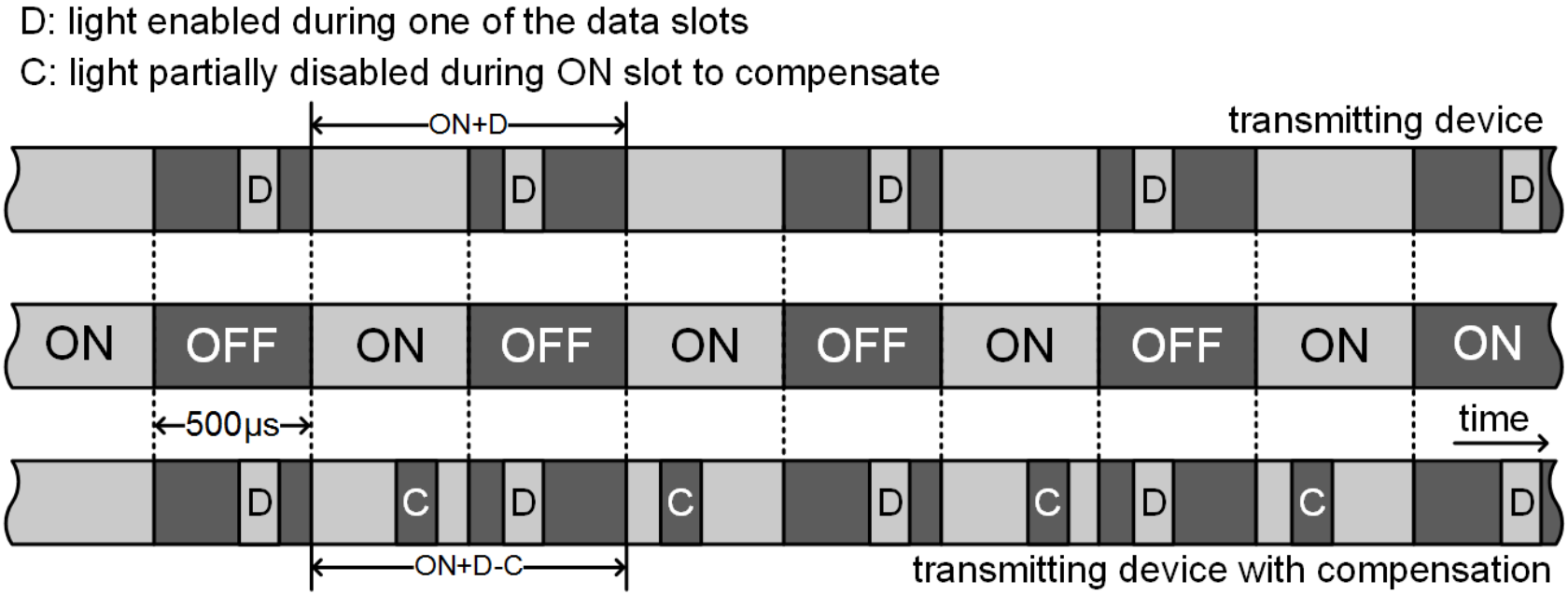
- Small synchronization slots (S1,S2) to reduce noise effects
- S1 and S2 are used to align (synchronize) the OFF slots
- Data intervals protected by guard intervals (G) to prevent light leakage (due to imprecise synchronization)

Data Encoding & Medium Sensing



- Data decoding without detection threshold
- Well defined medium busy/idle states
  - D1 and D2 differ significantly -> medium busy
  - D1 and D2 close -> medium idle

Constant Light Output



- Transmitting data adds additional light output -> visible brightness changes
- Compensation during following ON slot to keep average light output constant

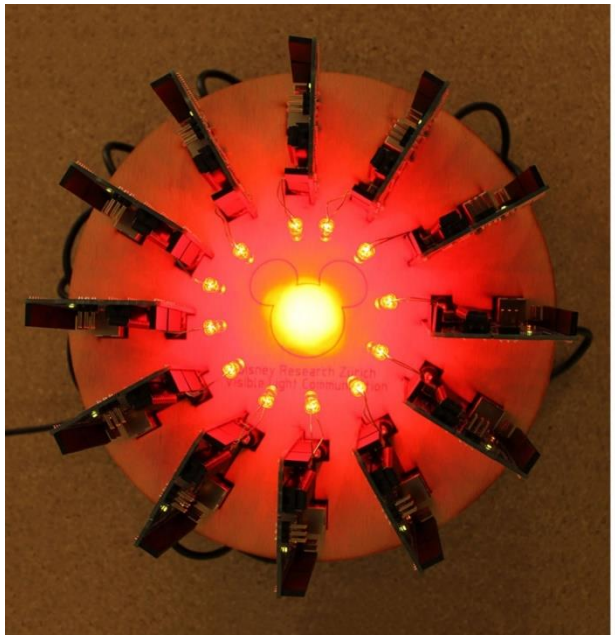
LED-to-LED Networks

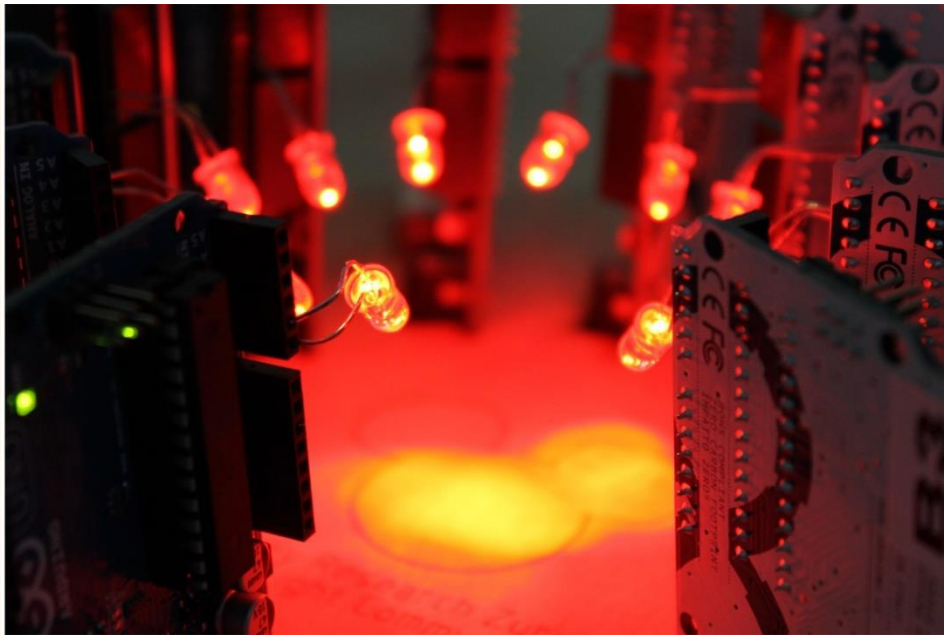
Physical Layer

- Illumination
- Synchronization
- Modulation / demodulation
- Forward Error Correction

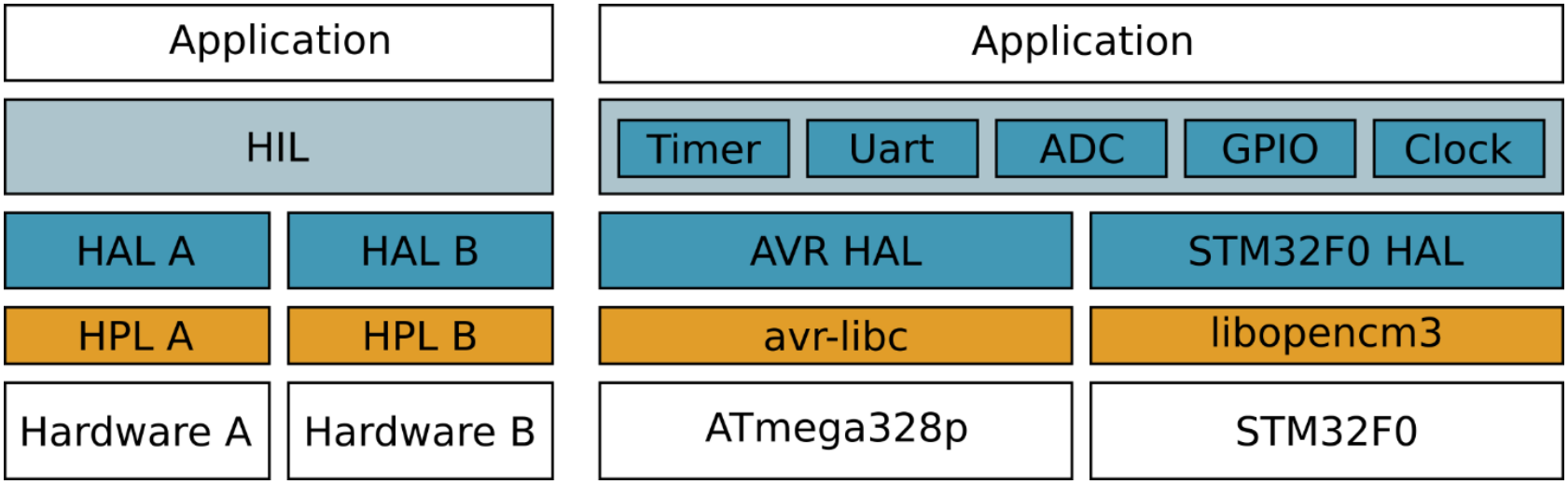
Medium Access Control Layer

- Listen before talk
- Contention-based backoff
- RTS/CTS



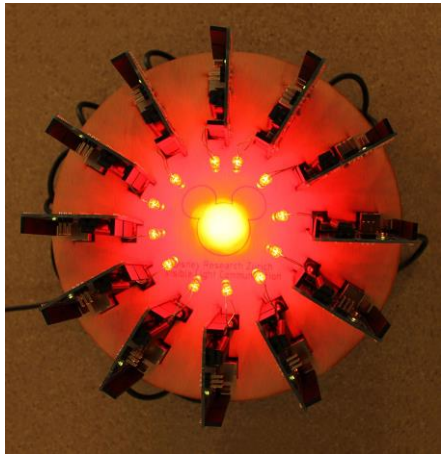


Portable Software Solution (HAA)

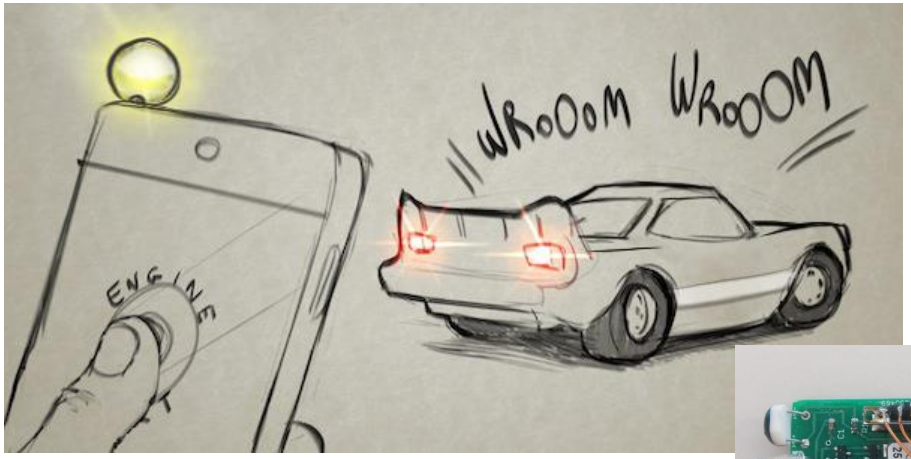


- HAA : Hardware Abstraction Architecture
- HIL : Hardware Interface Layer
- HAL : Hardware Adaption Layer
- HPL : Hardware Presentation Layer

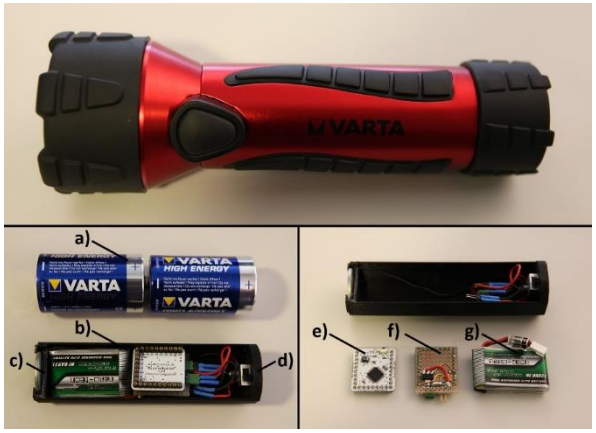
Building Blocks



LED-to-LED networking



From sound to sight



VLC-based flashlight

VLC-enabled light bulbs

Linux Light Bulbs

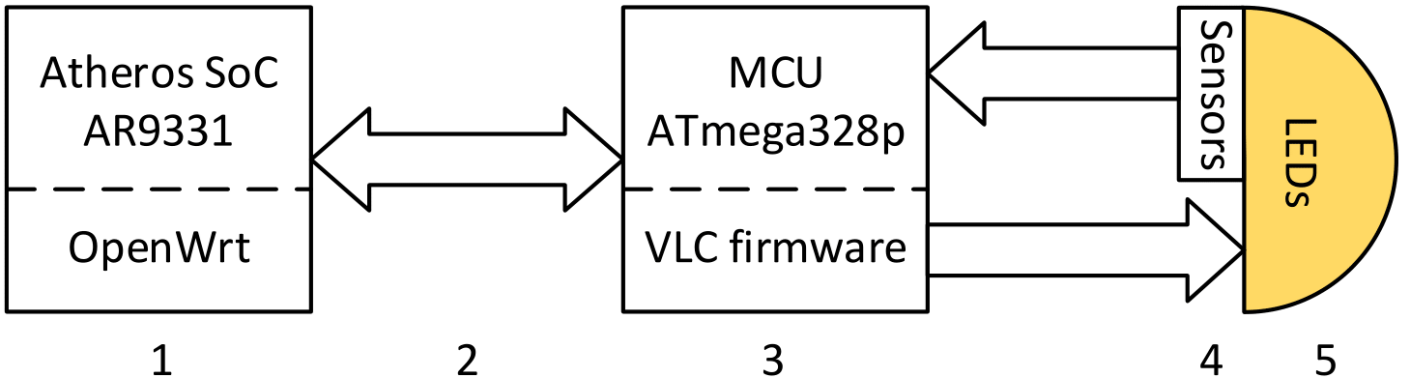
Motivation – Localization

- Positioning (devices, people)
  - Find your location
  - Location-based customized configuration
  - Trigger actions based on location
- ~~Solution: GPS~~ does not work for indoor scenarios
- Indoor localization is still a big issue…

Goal

- VLC-based "Internet of Things"
- Requires Internet Protocol (IP) connectivity
  - Light bulb ←→ smart device/toy
  - Light bulb ←→ light bulb
- Linux integration
  - Linux Network Stack → IP, TCP, UDP …
  - Reuse of existing protocols and applications
- Independent control channel
  - Measurements
  - Debugging and firmware updates

System Architecture



1) Wi-Fi–enabled System on a Chip (SoC)

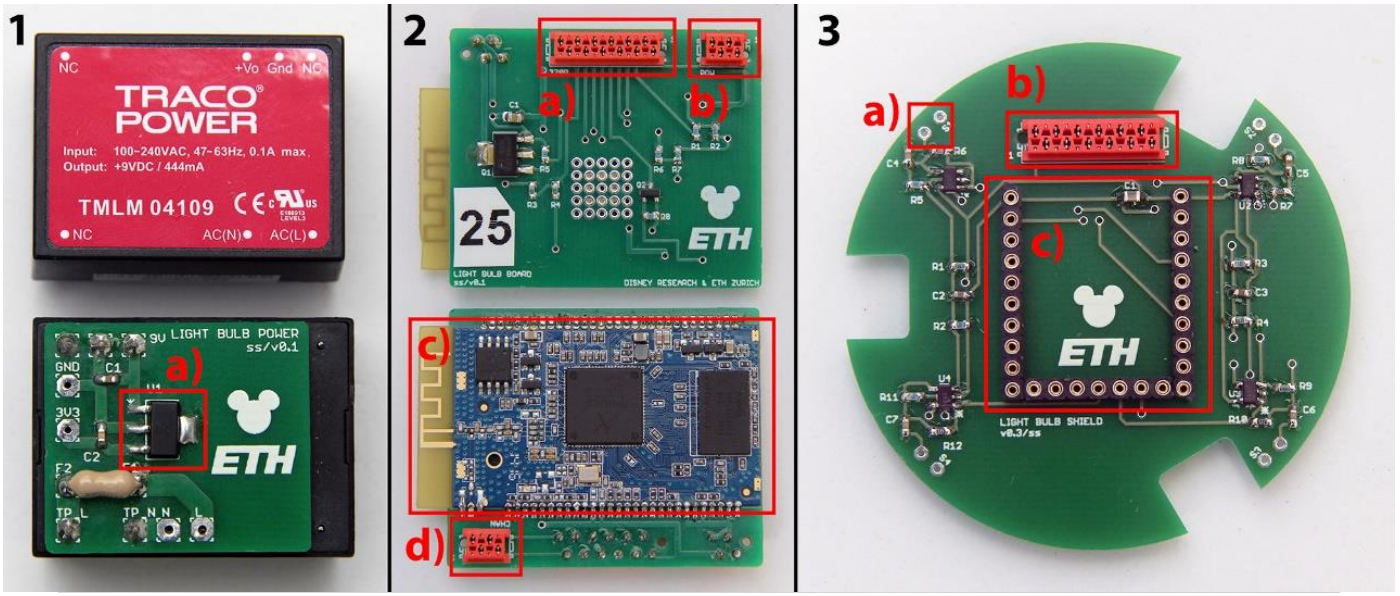
2) UART interface (serial)

3) Microcontroller running VLC firmware (PHY and MAC layer)

4) Receiving diode with amplifier

5) Original LED plate

System Components



1) Power supply with regulator

2) Qualcomm Atheros SoC running OpenWrt

3) VLC controller and sensor board

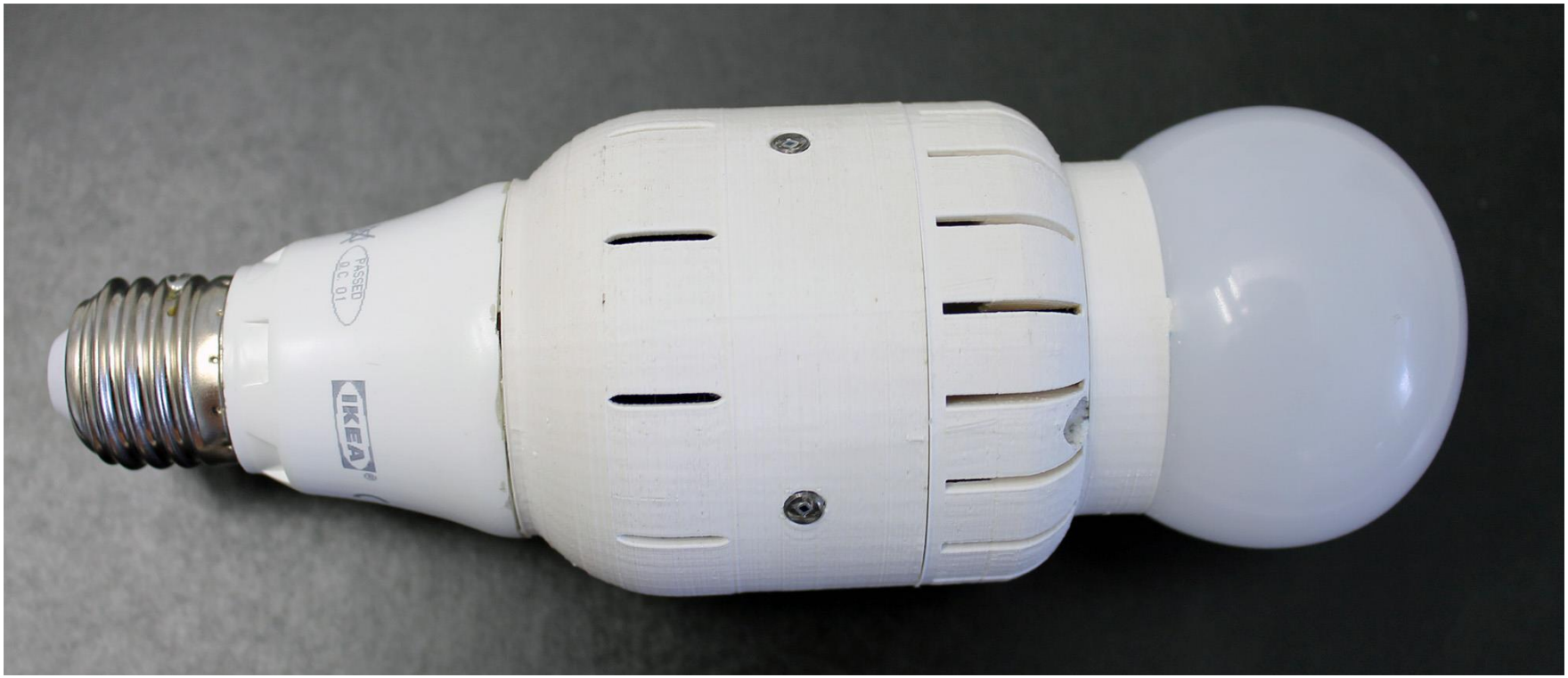
Light Bulb Casing

1) Bottom casing for additional electronics

2) Top casing for LED plate and heat sink

3) Original light bulb socket with power supply

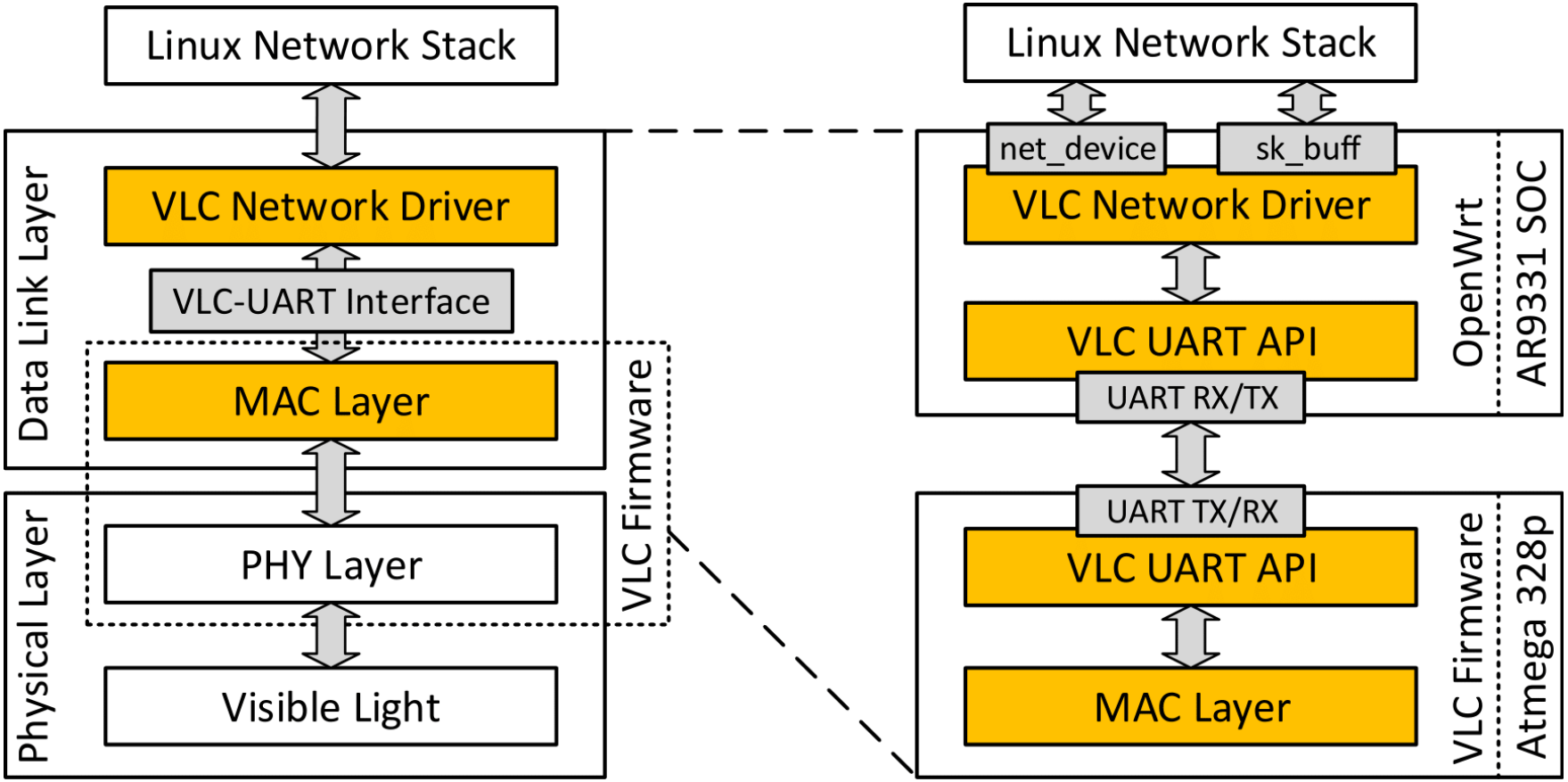
Fully Assembled Light Bulb



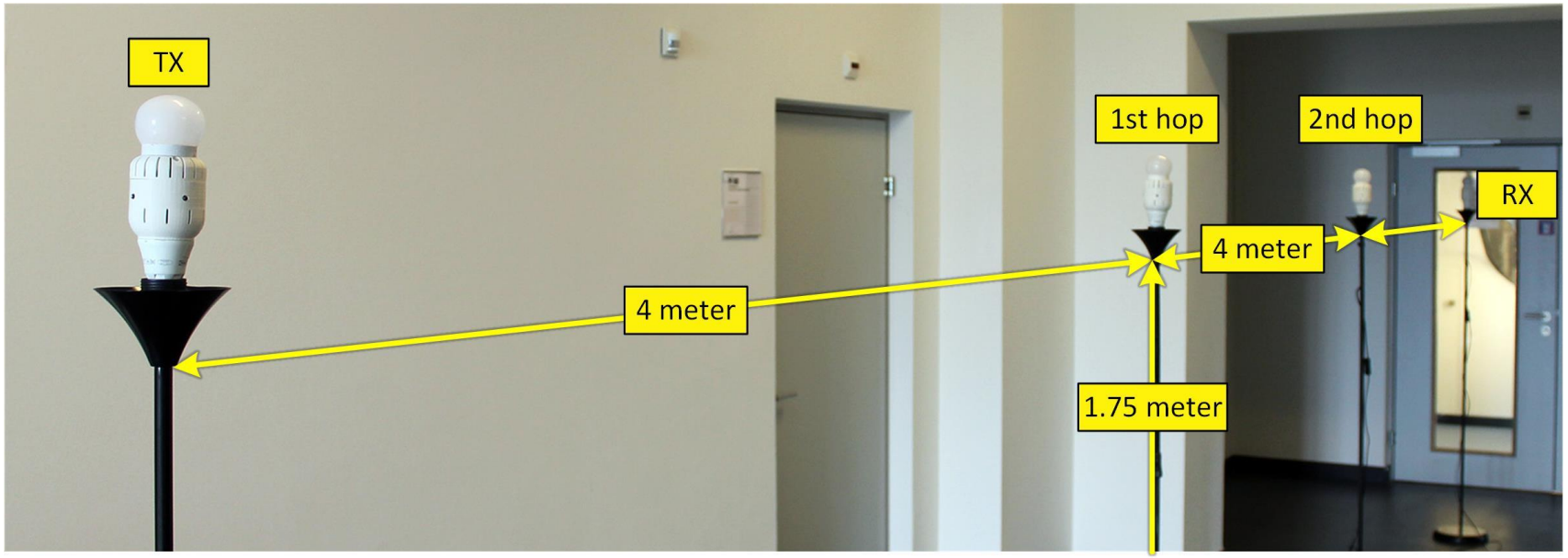
Linux Driver Implementation (1)

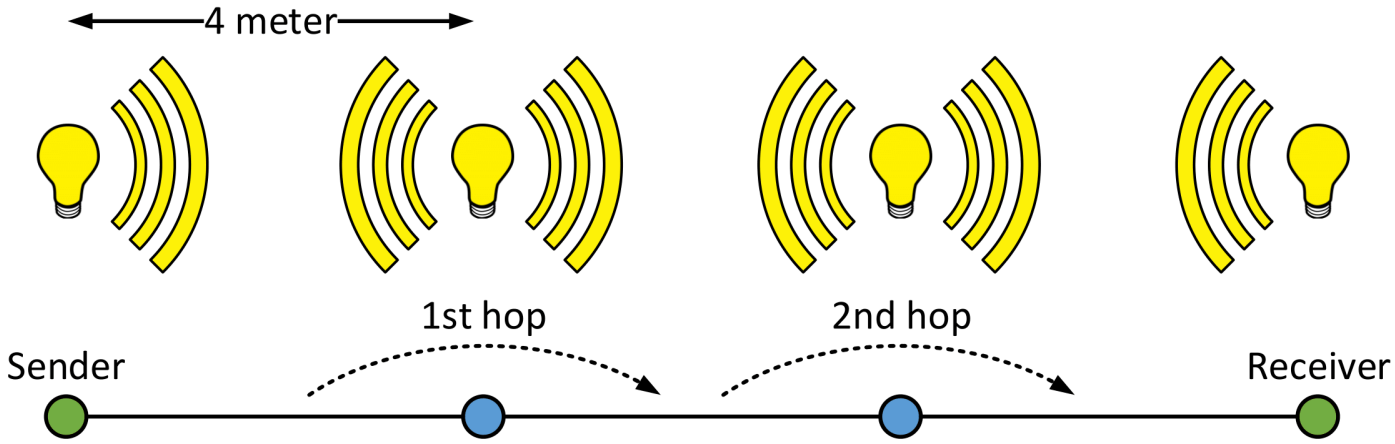
- VLC microcontroller as external device
  - Separate operating system from VLC hardware and firmware
  - UART (serial) used as communication interface
- Kernel module
  - Transparent integration with the Linux network stack
  - VLC microcontroller integrated as Ethernet device → vlc0
  - Driver and VLC microcontroller implement an API for data exchange and control
  - Linux network stack → driver → VLC microcontroller
  - VLC microcontroller → driver → Linux network stack
- Any IP based protocols or applications can be reused

Linux Driver Implementation (2)

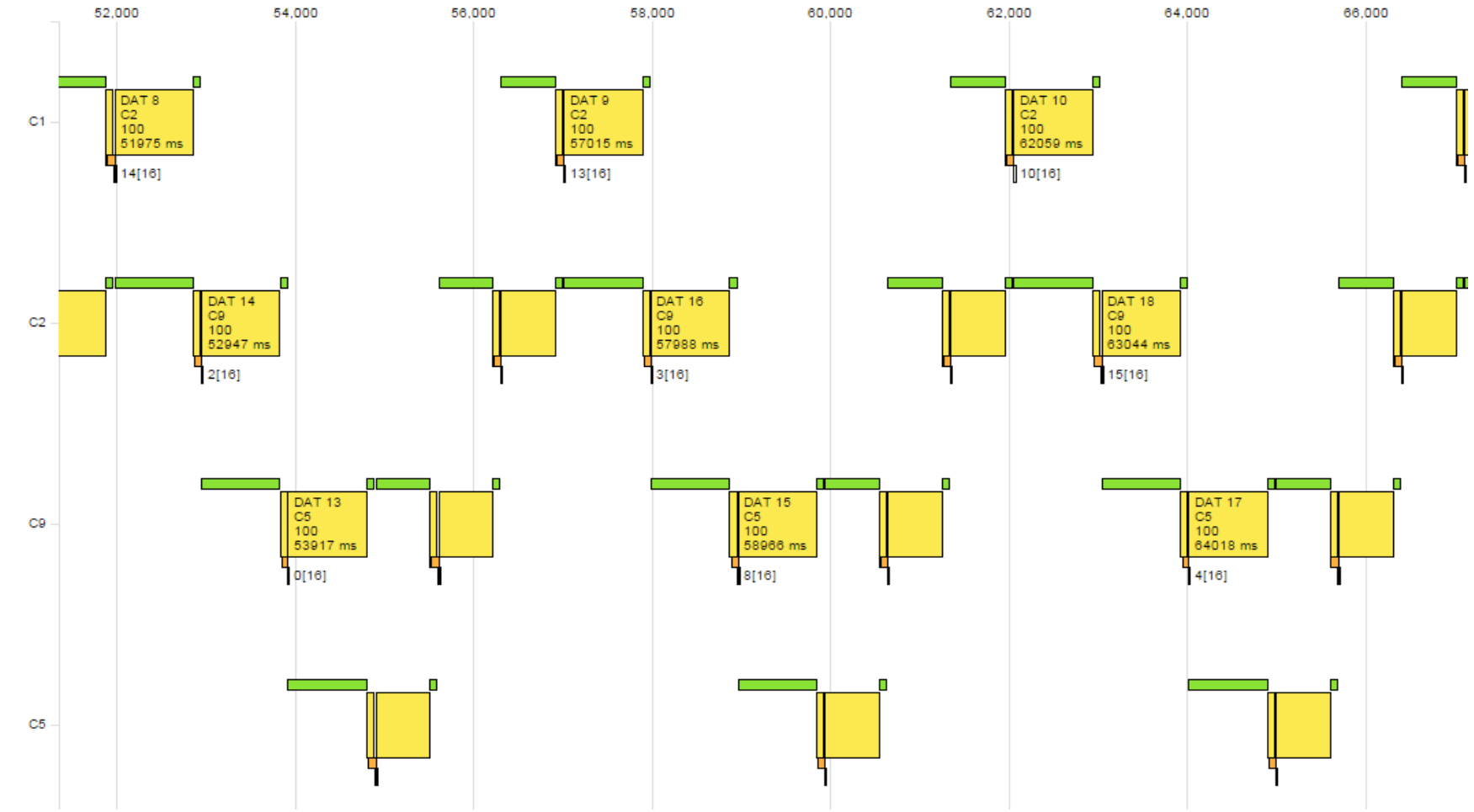


Testbed Setup





TCP Flow Visualization



Packet (ACK, DATA) Receiving Indicator

Conclusions

- Proof of concept IP-enabled VLC light bulbs
- Transparent Linux integration enables reuse of existing applications and protocols
- VLC-based IP multi-hop networking
  - Enables communication between light bulbs and smart objects
  - No additional radio technology necessary
- Technology enables:
  - VLC-based indoor localization
  - Lighting control systems
  - Internet of Toys